Amendments to the Specification

Please replace the paragraph on Page 4, lines 5 - 20 with the following marked-up replacement paragraph:

-- In a second prior art approach, sometimes referred to as an "natural key" approach, some set of one or more properties that uniquely identify a specific type of resource are designated as "key properties" or "identity properties". All access to instances of that type of resource must therefore specify the values of [[the]] each key property to identify the desired instance. A key disadvantage of this approach is that, in an OO system, only classes that have the same identity mechanism can be treated the same. Or, stated in OO terms, only classes with the same identity mechanism can be manipulated polymorphically. This creates a conflict when assigning key values. That is, to facilitate polymorphic access, keys should be defined near the root of the inheritance hierarchy in which the classes are defined. But, the values of the keys must be passed to the agents that will actually interact with the resource instances, and an agent must be able to use the identity it is passed (e.g., in a management request message) to find the target resource instance. This means that the keys must map to those properties in the real IT environment that identify a resource of a specific type. This requirement for identifying actual resources tends to force declaration of keys down to the leaf classes, or bottom, of the inheritance hierarchy. When key properties are located at the leaf classes, then classes between the leaves and the root are effectively unmanageable because there is no means to address class instances. --

Please replace the paragraph on Page 15, lines 4 - 10 with the following marked-up replacement paragraph:

-- For example, suppose a class "person" has one instance representing "Joe" and another instance representing "Mary". Mary may work for Alpha company, and have an employee identifier ("ID") that uniquely identifies her in within the scope of a class such as "Alpha_Employees". ~~Alternatively,~~ Joe might not work for this company, in which case he is not identified by one of its employee IDs. Instead, Joe might work for Beta company, and have a unique Beta company ID; or, Joe might be identified by properties that are completely different, such as by his name and the community in which he lives. --

Please replace the paragraph that begins on Page 32, line 20 and carries over to Page 33, line 9 with the following marked-up replacement paragraph:

-- If the test in Block 715 has a negative result, then this instance does not need a scoping context in its identity, and control therefore transfers directly to Block 725. Block 725 tests whether this identity is universally unique. (In one approach, "Element" class 110 of Fig. 1 may include a separate "RootScope" property with which each class may specify the root scope, if needed. The value for the root scope portion of the instance identity may be created, for example, by obtaining the class name of the class that provides the root scope for this instance, along with the property name/value pair(s) specified in the naming rule for that class.) If ~~class) If~~ the test in Block 725 has a positive result, then the processing of Fig. 7 is complete. Otherwise, Block 730 prepends a root scope to the instance's identity, after which the identity is guaranteed to be unique. The identity creation technique of Fig. 7 is then complete. --

Please replace the paragraph on Page 36, lines 9 - 17 with the following marked-up replacement

paragraph:

-- In an optional aspect, the identity string syntax described herein may further include a format identifier that is specified prior to the root scope. For example, the syntax "Tivoli:" (without the quotation marks) might be used. Use ~~used Use~~ of a format identifier is not required, but allows identity strings to more closely resemble industry-standard identifiers constructed according to naming schemes such as that specified by the Internet Engineering Task Force ("IETF") in its Request For Comments ("RFC") 2141 (May 1997), which is titled "URN Syntax". If a format identifier is used in an identity string, use of the colon symbol in the format identifier syntax allows easily distinguishing that portion of the identity string from the root scope, scoping context, and local identity. --

Please replace the paragraph that begins on Page 38, line 17 and carries over to Page 39, line 2 with the following marked-up replacement paragraph:

-- While the preferred embodiments of the present invention have been described, additional variations and modifications in those embodiments may occur to those skilled in the art once they learn of the basic inventive concepts. Therefore, it is intended that the appended claims shall be construed to include [[both]] the preferred ~~embodiment~~ embodiments and all such variations and modifications as fall within the spirit and scope of the invention. --